AD-A104 887 BATTELLE COLUMBUS LABS DURHAM NC F/G 12/2
BRIEF USABILITY SURVEY OF OPERATIONS RESEARCH APPLICATION SOFTW--ETC(U)
DEC 78 D YOUNG
UNCLASSIFIED NL

1 OF 1
AD
A104887

END
DATE
FILMED
10 81
DTIC

(15)

BRIEF USABILITY SURVEY OF OPERATIONS RESEARCH APPLICATIONS SOFTWARE FOR DECISION SUPPORT SYSTEMS

22 Dec[redacted]978

By DONOVAN YOUNG
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta GA 30332

Submitted to

Battelle Columbus Laboratories
Durham Operations
3333 Chapel Hill Boulevard
P.O. Box 8796
Durham NC 27707

In fulfillment of Scientific Services Agreement D.O. No. 0919

Contracting Officer's Technical Representative:

James M. Irwin
AIRMICS
Georgia Institute of Technology
Atlanta GA 30332

Findings and opinions expressed herein are those of the author and should not be construed as official Department of the Army positions, policies, or decisions, unless so designated by other documentation.

Security Classification: UNCLASSIFIED

Table cf Contents

SUMMARY

BRIEF USABILITY SURVEY OF OPERATIONS RESEARCH APPLICATIONS SOFTWARE FOR DECISION SUPPORT SYSTEMS

By DONOVAN YOUNG

28 December 1978

(Unclassified)

This report proposes and applies a ranking procedure for identifying operations research models having promising characteristics for use in small minicomputer-based decision support systems of greatest potential value in Army decision contexts. An ideal profile of attributes is suggested. A multivariate statistical classification is performed to evaluate actual attributes of a comprehensive list of available models against the ideal profile. It is concluded that the most promising operations research models for minicomputer-based Army decision systems include an interactive minimax location algorithm, a generalized decision tree model, some inventory models, and PERT/CPM planning. Shortcomings in the data are pointed out, especially with regard to estimating Army needs for implementations of various models. A computer program is furnished to allow convenient further experimentation in identifying promising models.

BRIEF USABILITY SURVEY OF OPERATIONS RESEARCH APPLICATIONS SOFTWARE FOR DECISION SUPPORT SYSTEMS

By DONOVAN YOUNG

1. INTRODUCTION

The Management Information Science Division of AIRMICS has begun to develop a research interest in potential Army applications of minicomputer-based decision support systems. Part of that interest concerns real-time interactive use of operations research applications models. It was decided to survey operations research applications models as to their usability in minicomputer-based decision support systems for Army use, and this report presents the results of the survey.

As given in the Statement of Work for this survey (Appendix A), the objective is "to provide an evaluative classification of operations research applications software . . . as to their demonstrated and potential usefulness in minicomputer-based decision support systems." Specific tasks include classifying software as to characteristics affecting use in decision support systems, reviewing the interactive-computing experience of each class, recommending a procedure for selecting the most promising models, and recommending a small set of operations research algorithms for possible pilot implementation.

2. DEFINITIONS

The population of operations research models to be surveyed is given in the work statement (Appendix A) as "encompassing optimization, simulation and queueing." This was not intended to be a narrowly

restrictive definition, and it was decided to include in the survey applications software for all algorithms and procedures that are regularly discussed in three leading journals and in the current editions of four leading textbooks:

...Operations Research

...Management Science

...Mathematics of Operations Research

...Taha, Operations Research (Macmillan)

...Wagner, Principles of Operations Research (Prentice-Hall)

...Hillier and Lieberman, Introduction to Operations Research (Holden-Day)

...Phillips, Ravindran and Solberg, Operations Research Principles and Practice (Wiley)

In particular, the operations research coverage extends not only to optimization, simulation, and queueing, but also to those aspects of several technical fields that do not necessarily involve optimization, simulation, or queueing. These fields include project scheduling (PERT and CPM), inventory control, decision theory, game theory, Markov processes, and other stochastic processes where applied as decision aids.

On the other hand, several fields are excluded even though they are often associated with operations research, since most of their literature lies outside the literature of operations research. These excluded fields (together with keywords for their central literature where different from the name of the field itself) are logistics, facility layout and facility location (industrial engineering), reliability and quality control (statistics), forecasting (statistics), time series analysis (statistics),

cluster analysis or numerical taxonomy (statistics), multiple regression (statistics), job shop scheduling (operations management), assembly line balancing (operations management), value theory (systems engineering), control theory (systems engineering), linear systems theory (electrical engineering), econometrics models, capital budgeting (finance), and engineering economy models—except, in most cases, where the models primarily involve optimization, simulation or queueing per se. Certain models definitely lie outside operations research even though they are specifically optimization models; these are least-squares methods in statistics (including maximum-entropy formalization) and in numerical analysis (e.g. matrix inversion and numerical integration), input-output models in economics, and models confined to the literatures of actuarial science, computer science, marketing, chemistry, physics, molecular biology, and all other fields having literatures whose overlap with the literature of operations research is minimal.

The population of decision support systems to be considered is given in the work statement (Appendix A) as "minicomputer-based decision support systems," where the term "decision support system" (DSS) is stated to include systems that provide "interactive response in interfacing with a data base to aid a non-programming individual in unstructured problem solving." A minicomputer-based DSS is a DSS whose chief hardware resource is a minicomputer system, usually with advanced graphics. Interactive response on a system that is minicomputer-based is taken to imply real-time rather than time-sharing operation.*

*This distinction turns out to have no effect on model selection.

There are two publications generally considered to be most authoritative in defining DSS. These are the very broad taxonomic article by Steven L. Alter ("A Taxonomy of Decision Support Systems," Sloan Management Review, Fall 1977, pp41-42), and the 1978 book by Peter G. W. Keen and Michael S. Scott Morton (Decision Support Systems, Addison-Wesley). In Alter's taxonomy, the DSS of interest here are those that are model-oriented, as opposed to data-oriented, both because minicomputer implementation excludes large data bases and because this survey focuses on models. Within the Keen/Scott-Morton set of case studies, the DSS of interest here are typical, except that the large data-oriented systems (those that primarily provide ad-hoc statistical analysis and report generation from large data bases) and the geodata-oriented "war-room" systems are outside the scope of this survey.

As compared with the bulk of DSS reported in the literature, the DSS of interest in this survey are more oriented toward middle management than toward the kind of top-level corporate-planning DSS that have received the greatest publicity. AIRMICS contemplates research in DSS that can be fielded at multiple locations, not highly-classified Pentagon-level systems.

In this survey, DSS will hereafter mean a model-oriented minicomputer system, usually with advanced graphics, expressly designed to support unstructured real-time interactive problem solving by a non-programming individual. "Model-oriented" implies that the computation load is relatively more significant than the data management load. "Unstructured" problem solving means "experimental" or "what-if" or "cut-and-try" problem solving, in which there exist junctures at which

the user actively participates, entering data or commands that are influenced by the previous outcomes.

Operations research terminology sometimes confusingly overlaps computer terminology. In particular, "programming" is used in the sense of "planning" and may be taken as equivalent to "optimizing." "Algorithm" quite narrowly denotes a procedure that guarantees an optimal solution to a "program" or optimization problem; a "heuristic" is a procedure that gives a good but not necessarily optimal solution. A "code" is software that implements a procedure. "Model" means a problem, the mathematical formulation of a problem, a procedure, or even the "code" that implements a procedure, but properly it denotes a framework for formulation of a set of similar problems. Models may be prescriptive or descriptive, and a given model often may be expressed in several ways—for example, on paper, as data, or as logic structure of a computer program.

In this survey, model will hereafter mean a specific framework for formulation of a set of operations research problems that share a similar structure. "Operations research" problems are those treated in the operations research literature, as clarified at the beginning of this section.

With these definitions, the purpose of this survey is to find the models best suited for DSS implementation within the Army.

3. SURVEY METHOD

There exists a consensus taxonomy of operations research models that may be derived from comparing the taxonomies implied by literature indices, session-title lists from recent professional meetings, departmental editorship lists from professional journals, and textbook section

headings. This list is given in the following section. Each entity in the list is a model or set of closely similar models.

Each entity in the list of models is viewed as having a vector of attributes—more properly, a vector of variables, each measuring the degree or amount of a given attribute. For example, linear programming models are very general, whereas dynamic programming models are highly problem-specific; thus linear programming models would have high values of the attribute "generality" whereas dynamic programming models would have low values of the same attribute. As another example, event simulation models have had much military usage, feedback-dynamics models little; thus the attribute "history of successful military application" would have a high value for the first, a low value for the second.

The same attributes may be related to DSS, and a multivariate statistical technique can be applied. If the goal were to derive groups of models that would behave similarly with respect to DSS-related attributes, then cluster analysis would be the applicable technique; for example, if one model were known to have been very successfully used in a DSS, the cluster analysis results would indicate which other models were most similar with respect to the attributes, suggesting that these similar models would be good candidates. (Also, the cluster analysis results would generate a taxonomy, perhaps much different from the listed taxonomy, that would make maximal sense in discussing DSS-related issues.) On the other hand, if data on a group of existing "good" and "bad" DSS were available, discriminant analysis could be applied to the data to derive an ideal profile of attributes for a model to have, and in the second or "classification" stage of discriminant analysis the models in the list could be ranked according to their statistical distance

from this profile. The data on existing DSS could alternatively be applied in <u>factor analysis</u> to eliminate irrelevant attributes and to rearrange the remaining attributes into factors (groups of attributes) that would simplify the classification problem.

Clearly the applicable method for this survey includes as its final step the <u>statistical classification of models according to the closeness of their vectors of attributes to an ideal vector of attributes</u>. Given the ideal vector, the calculations are quite simple: If there are N different attributes, numbered from 1 to N, and if a_{ij} is the value of the ith attribute of the jth model and b_i is the value of the ith attribute of the ideal model, then the unweighted distance for the jth model is

$$d_j = \sum_{i=1}^{N} (a_{ij} - b_i)^2$$

Assuming the attributes are all similarly scaled (for example, if all attribute values range from 0 to 1), and assuming each attribute is considered equally important, the unweighted distance d_j can be taken as a direct ranking variable for the jth model. Differences in importance among attributes may be taken into account by defining a set of weights $\{w_i\}$ so that the weighted distance $d_j^{(w)}$ is defined as

$$d_j^{(w)} = \sum_{i=1}^{N} w_i^2 (a_{ij} - b_i)^2$$

In either the unweighted or the weighted case, a distance of 0 indicates a model whose attributes are exactly those of the ideal model, and increasing distances represent increasing departures from the ideal.

For detailed discussions and derivations of these distance metrics, the 1971 multivariate statistics textbooks by Cooley and Lohnes and by Tatsuoka are good references. For motivations and tutorial explanations, good references are the 1975 cluster analysis textbooks by Anderberg and by Sneath and Sokol, or, for very brief explanations with examples, the SPSS manual (Statistics Package for the Social Sciences, Second Edition, by Norman H. Nie et al, McGraw-Hill, 1975).

Since no data on evaluated model-oriented DSS are available, there is no empirical basis for establishing an ideal attribute profile. However, for each attribute, there exists either a theoretical or an intuitive basis for establishing an ideal value for that attribute. For example, data requirements should be as small as possible, successful prior experience should be as extensive as possible, etc. As each attribute is defined, its ideal value will be established.

The method of this survey will be as follows:

1. To establish a comprehensive list of operations research models (Section 4)
2. To define DSS-relevant attributes and establish their ideal values (Section 5)
3. To assign weights to all attributes and attributes to all models (Section 6)
4. To calculate distance metrics to all models and rank the models (Section 7)

The results will be used to derive recommendations for pilot implementation of those models that best fit DSS requirements.

4. TAXONOMY OF OPERATIONS RESEARCH MODELS

This is a taxonomy of models rather than software products, for several reasons:

1. Models are well covered in the open literature, products are not
2. Interest here is on possibilities, not history
3. Many actual products run only on large computers and do not use graphics, but the model they implement may be capable of DSS implementation nevertheless
4. The characteristics of a model, in combination with the characteristics of the DSS, provide a basis for estimating the characteristics of DSS implementation of the model.

Not all operations research software is applications software. An example is the GASP simulation language, which is simply a set of preprogrammed subroutines that makes a programmer's job easier; another example is SIMSCRIPT, a simulation language in which to do programming. On the other hand, such simulation languages as HOCUS and (marginally) GPSS do not actually require the user to be what would ordinarily be called a "programmer" and thus are of interest (and are considered applications software for the purposes of this survey).

The list to follow expresses a consensus taxonomy of operations research models that is largely a blend of two organizing principles. The stronger of the two is classification by model structure—especially among optimization models, those sharing similar mathematical characteristics are put together. Problem area is the other organizing principle. As examples, by structure we have fields such as linear programming and simulation, but by problem area we have fields such as project scheduling and inventory control (both of which use several structural models and hence refuse to fit into a single structural category).

Although the list is of models rather than software products, some models are inseparable from the software products implementing them, and are known by the product name.

Operations Research Models

Linear programming (general)

1. Production LP (APEX-III, MPSX)
2. Core-resident LP (MPOS)
3. Tutorial LP (EZLP, LEARNLP)
4. LP data generators and drivers (application-specific 'front-end' programs)
5. Small LP-centered applications programs
6. Two-player game model

Integer programming (general linear programming with integer constraints)

7. Mixed-integer branch-and-bound (Dakin's algorithm)
8. Mixed-integer cutting plane (Gomory's algorithm)
9. 0-1 integer (direct search)
10. Flow-shop algorithm (Baker's)
11. Knapsack algorithm (modified Bellman's)

Network flow (linear) programming

12. Assignment algorithm (Hungarian)
13. Transportation model (Hitchcock)
14. Out-of-kilter algorithm
15. Primal network flow algorithm
16. Dual shortest-path algorithm
17. Generalized network programming (NETG)

Nonlinear programming

18. Direct search algorithms
19. Cyclic coordinate algorithms
20. Steepest ascent algorithms
21. Second-derivative algorithms (Davidon-Fletcher-Powell)
22. Quadratic programming (Wolfe's, Beale's)
23. Geometric programming (partial condensation)

Project management

24. Production PERT/CPM with leveling and crashing (PAC-II)
25. Production PERT/CPM with graphics
26. Simulation-based PERT

Dynamic programming

27. Deterministic (general, one-state-variable, discrete)
28. Policy iteration (Markov decision process)
29. Production scheduling (deterministic)
30. Decision tree models
31. Generalized decision tree model
32. Sequential games

Inventory models

33. Deterministic lot-size (EOQ) models
34. Lot-size—reorder-point models
35. Periodic-review models
36. Dynamic lot-size models
37. 'Newsboy problem'

Simulation languages (usable at non-programming level)

38. HOCUS (small-scale event simulations)

39. GPSS (event simulation)

40. DYNAMO (feedback dynamics)

Interactive optimization and heuristics

41. Multicommodity flow heuristic

42. Traveling salesman heuristic

43. Minimax location interactive algorithm with arbitrary constraints in the plane

Queueing

44. Markov queue steady-state (MMQ)

45. Runge-Kutta transient M/M solution

5. ATTRIBUTES OF MODELS RELEVANT TO DSS

For a given model to be useful in DSS it must have appropriate attributes in three main categories: First, there must be a need in the Army for routine interactive solution of at least one of the problems the model can solve. Second, the model must be implementable as a DSS module on a minicomputer-based system as described in Section 2. Third, there must be a benefit for DSS solution of the problem as opposed to alternative (including existing) procedures.

Army need will be a single attribute estimated on the basis of existing Army use, expressed needs known to the author, or potential need derived from informal analysis of Army activities. Attributes pertaining to implementability are those that measure demands on resources, both during development and in actual use. Attributes pertaining to DSS benefit are those that measure the advantage of DSS solution as compared to other methods.

An ideally useful model would be one that solves a high-need problem, can easily be implemented on a minicomputer-based DSS, and has important advantages over "batch" solution of the same problem.

All attributes are arbitrarily scaled from 0 to 10. The scale is continuous, but in practice there is no need to use fractional values. For attributes that have units of frequency or probability, the attribute value is 10 times the frequency or probability; for attributes that are yes/no (binary), the end-points of 0 and 10 are used.

The following list gives the attributes used in this study:

Model Attributes

No.	Name and description	Ideal value
1.	Army need. Probability that at least one of the problems solvable by the model is needed for Army decision making on a routine, multisite basis (×10). 10 for current use.	10
2.	Storage demand. 10 for models and data too large to run on minicomputer-based DSS as described in Section 2, 0 otherwise. Binary, not continuous.	0
3.	Computation load. 10 for computation load making solution times too long for response and turnaround appropriate to the model. Proportional to reported loads.	0
4.	Development effort. From 0 if applicable code and documentation exists, to 10 if DSS implementation would be major conversion effort.	0
5.	Interactive response time. Number of seconds of wait to be experienced at interactive epochs during solution, after correction for wait masking, etc. (10 sec bothers user)	0
6.	Data demand. Subjective measure of difficulty of providing data for running the model. 10 for 'data-heavy' models.	0
7.	Turnaround advantage. Subjective measure of advantage in getting quick solution to model, as opposed to batch turnaround.	10

8. Interactivity advantage. Subjective measure of advantage (but in addition to turnaround advantage) in solution quality or usefulness due to user participation. 10

9. Intuitive advantage. Subjective measure of advantage (but in addition to attributes 7 and 8) due to better user understanding gained by participating in solution. 10

As can be seen from the ideal values, the best possible operations research model to implement in a pilot DSS would be one that solves a problem already routinely solved (1), has insignificant storage and computer-time demands (2 and 3), needs no development effort (4), has instantaneous response (5), demands no data (6), greatly benefits the user by giving quick turnaround, (7) gives a much better or more useful solution as a result of user participation (8), and allows the user to understand the problem much better (9).

6. ESTIMATES OF ATTRIBUTES

The Army need attribute is 10 for those models already having multisite routine use, including the assignment algorithm and probably the transportation model, production PERT/CPM with leveling and crashing, decision tree models, deterministic lot-size (EOQ) models, probably lot-size—reorder-point models, probably periodic-review models, and probably GPSS. It is 8 for those models that undoubtedly justify multisite routine use except for unavailability of implementation, including (probably several) small LP-centered applications models, all the network flow models, all PERT models, generalized decision trees, all inventory models, HOCUS, and minimax location. It is 6 for all the rest, reflecting a judgement that each one has a probability of roughly 60% of being useful in several places if it were implemented in a DSS and made available in a suitable application-

decision-aid facility.

The storage demand attribute is 10 for those models that have no reasonable hope of running satisfactorily on a minicomputer-based DSS because of storage demand; the attribute is 0 for models posing no storage difficulty. Core requirements are usually given directly in the literature on large-scale models, because often the storage of intermediate results and storage of the code itself are significant. The models having a storage demand attribute of 10 are production LP, all the integer programming algorithms, and policy iteration. All the other models (when applied to sufficiently small problems) are assumed to fit and are given a value of 0 for the storage demand attribute. The applications considered here are thus smaller than ordinary with respect to core-resident LP, LP data generators and drivers, generalized networks, and GPSS. The multicommodity flow heuristic and the traveling salesman heuristic are given a value of 5 since they may or may not fit when applied to reasonably-sized problems.

The computation load attribute would be 10 for those algorithms whose computation times would intrinsically be incompatible with interactive response-time requirements; despite some notorious number-crunchers among the listed algorithms, none are intrinsically incompatible with DSS when applied to suitable problems. All the integer and nonlinear programming algorithms are assigned a 3 to reflect their generally long computation times; simulation-based PERT is assigned a 2; GPSS is assigned a 3, and DYNAMO is assigned a 1. All the other models are assigned 0.

The development effort attribute is 10 for models whose conversion to DSS would be difficult, or whose only implementation is proprietary or poorly documented; these include production LP, LP data generators and drivers, Baker's flow-shop algorithm, generalized network programming, HOCUS, GPSS, and the traveling salesman heuristic. Some conversions would be moderately difficult and are assigned a 5; these include mixed-integer algorithms, the primal network flow algorithm, all the project management packages, DYNAMO, and the multicommodity flow heuristic. Other models have a 0 development-effort attribute.

The interactive response time attribute is very highly correlated with the computation-load attribute, but was included in order to distinguish those cases in which outputting of intermediate results or breaking the solution into steps could provide acceptable response times despite high computation load, and conversely those cases in which many response cycles represent only one actual solution. This attribute is assigned the same values as the computation-load attribute, with the following exceptions: The response-time attribute is 0 for nonlinear programming algorithms 18, 19, 20 and 21, because intermediate reporting is beneficial inasmuch as the algorithms often give an acceptable solution in a short time and use most of the time creeping ever more slowly towards a more accurate solution; the attribute for simulation-based PERT is 0 because intermediate reporting is beneficial inasmuch as a small sample may give a satisfactory answer.

The data demand attribute is 10 for data-heavy models, which include only production LP and core-resident LP (although of course the data may be heavy for certain applications of other optimization models) and for GPSS, since here the 'data' includes the model structure logic.

The turnaround advantage attribute measures the advantage of getting an immediate solution interactively as opposed to a batch solution. This attribute is 8 for all the models in which the data are likely to be significantly arbitrary and subject to immediate change by the decision maker in response to the output (that is, where the problem is essentially a what-if problem); these models include the two-player game, the redirectable nonlinear programs of models 18 through 21, the project management models, decision trees, generalized decision trees, sequential games, all the simulation languages or models, and the queueing models. It is 5 for those models in which the basic data are likely to be fixed but there is likely to be a sensitivity-analysis step in which the decision maker can usefully supply arbitrary additional or revised data; this turns out to include every model in the list. The attribute is 10 for the interactive models, which require immediate turnaround.

The interactivity advantage attribute measures the advantage of obtaining better or more useful solutions because of interactive participation by the user. This is in addition to the turnaround advantage, which reflects the interactive effect on user time and convenience for quick turnaround. It is assumed that users will not trade solution quality for convenience, that is, they would make the required number of batch runs if DSS implementation were not available. With this assumption, the only solution quality advantage is for interactive procedures (models 41, 42, and 43). For these models the interactivity advantage attribute is 10, for others 0.

The intuitive advantage attribute measures the advantage gained by better user understanding of the problem structure or behavior when participating in the solution or redirecting the solution or seeing intermediate results or simply seeing results without delay so that the data that produced the results is still fresh in the user's mind. This advantage is strongest (10) for models in which the user can see results in graphical form; these models include the project management models, the decision tree and generalized decision tree models, the inventory models and the interactive models. The attribute is assigned a value of 8 for geometric programming and transient queueing, since interactive solution of these models strongly develops intuition about the problems modeled. The ease of sensitivity analysis contributes to learning for all other models to an extent of about 3 on the intuitive advantage scale. The attribute is assigned a 5 for tutorial linear programming since in that model learning about the model structure is the main goal.

These attributes will be referenced by brief one-word descriptors:

1.	Need	=	Army need
2.	Storage	=	Storage demand
3.	Load	=	Computation load
4.	Effort	=	Development effort
5.	Response	=	Interactive response time
6.	Data	=	Data demand
7.	Turnaround	=	Turnaround advantage
8.	Quality	=	Interactivity advantage
9	Learning	=	Intuitive advantage

The values of all attributes as set above are summarized in the following table:

Table of Attributes

Models	Attributes	Need 1	Storage 2	Load 3	Effort 4	Response 5	Data 6	Turnaround 7	Quality 8	Learning 9
Production LP	1	6	10	0	10	0	10	5	0	3
Core-resident LP	2	6	0	0	0	0	10	5	0	3
Tutorial LP	3	6	0	0	0	0	0	5	0	5
LP data generators	4	6	10	0	10	0	0	5	0	3
Small LP applications	5	8	0	0	0	0	0	5	0	3
Two-player game	6	6	0	0	0	0	0	8	0	3
Dakin's mixed integer	7	6	10	3	5	3	0	5	0	3
Gomory's mixed integer	8	6	10	3	5	3	0	5	0	3
0-1 integer	9	6	10	3	0	3	0	5	0	3
Baker's flow-shop	10	6	10	3	10	3	0	5	0	3
Knapsack	11	6	10	3	0	3	0	5	0	3
Assignment algorithm	12	10	0	0	0	0	0	5	0	3
Transportation model	13	10	0	0	0	0	0	5	0	3
Out-of-kilter	14	8	0	0	0	0	0	5	0	3
Primal network flow	15	8	0	0	5	0	0	5	0	3
Dual shortest-path	16	8	0	0	0	0	0	5	0	3
Generalized network	17	8	0	0	10	0	0	5	0	3
Direct search	18	6	0	3	0	0	0	8	0	3
Cyclic coordinates	19	6	0	3	0	0	0	8	0	3
Steepest ascent	20	6	0	3	0	0	0	8	0	3
Davidon-Fletcher-Powell	21	6	0	3	0	0	0	8	0	3
Quadratic programming	22	6	0	3	0	3	0	5	0	3
Geometric programming	23	6	0	3	0	3	0	5	0	8
PERT PAC-II	24	10	0	0	5	0	0	8	0	10
PERT with graphics	25	8	0	0	5	0	0	8	0	10
Simulation PERT	26	8	0	3	5	0	0	8	0	10
General dynamic prog	27	6	0	0	0	0	0	5	0	3
Policy iteration	28	6	10	0	0	0	0	5	0	3
Production scheduling	29	6	0	0	0	0	0	5	0	3
Decision trees	30	10	0	0	0	0	0	8	0	10
Generalized decision tr	31	8	0	0	0	0	0	8	0	10
Sequential games	32	6	0	0	0	0	0	8	0	3
EOQ models	33	10	0	0	0	0	0	5	0	10
Lot-size—reorder point	34	10	0	0	0	0	0	5	0	10
Periodic-review	35	10	0	0	0	0	0	5	0	10
Dynamic lot-size	36	8	0	0	0	0	0	5	0	10
Newsboy problem	37	8	0	0	0	0	0	5	0	10
Small-scale simulation	38	8	0	0	10	0	0	8	0	3
GPSS	39	10	0	3	10	3	10	8	0	3

DYNAMO	40	6	0	1	5	1	0	8	0	3
Multicommodity flow	41	6	5	0	5	0	0	10	10	10
Traveling salesman	42	6	5	0	10	0	0	10	10	10
Minimax location	43	8	0	0	0	0	0	10	10	10
Steady-state queue	44	6	0	0	0	0	0	8	0	3
Transient queue	45	6	0	0	0	0	0	8	0	8

As is customary in choosing and defining variables for multivariate analysis, the attributes are defined so as to be roughly equally important. However, given the estimated values, not all variables have the same range (maximum minus minimum). The nine attributes here have ranges of estimated values equal to 4, 10, 3, 10, 3, 10, 5, 10, and 7, respectively. Variables having smaller or greater ranges have less or more influence in determining distance (see the unweighted distance equation on page 7). This alone does not necessarily justify weighting, since if the entire population varies little with respect to some attribute (here, for example, load and response have narrow ranges), then it is proper for that attribute to have little influence.

However, it is often desired to study, and perhaps change, influences of various attributes. This is accomplished by setting weights to values greater or less that 1.0 in the weighted distance equation (page 7 and line 720, Appendix B). In particular, to increase the influence of the ith attribute by a factor of approximately F, its weight is multiplied by $\sqrt{F}$; and if M variables are to have approximately the same influence, grouped together, as each one had previously, their weights are each multiplied by $1/\sqrt{M}$.

The attribute values used in the present survey are not sufficiently well founded to justify weighting them a priori.

7. CALCULATIONS AND RESULTS

The simple computer program listed in Appendix B was utilized to calculate distances for various weightings according to the method given in Section 3. The program allows the user to specify weights, and it calculates and reports the distance metric for each model.

The results of two runs, one unweighted and one weighted, are reported here. Printed output of the runs is given on page 22.

The unweighted run gives the smallest distance metric for model 43, a minimax location algorithm that was designed especially for DSS implementation. Other high-ranking models are model 41 (a multicommodity flow heuristic also designed for DSS), model 30 (decision trees), and model 31 (generalized decision trees). Relatively high-ranking models are models 33, 34 and 35 (inventory models), 36 and 37 (other inventory models) and model 24 (PERT/CPM).

It can be seen that in the unweighted calculations the need attribute has very little influence (note the small correlation between need and the distance metric), and that the last three attributes, all measuring advantages of DSS implementation, have very great influence.

It may be reasoned that Army need should be an important factor in choosing a model, and that the three measures of DSS advantage should carry less weight, perhaps the weight of one measure rather than three. Accordingly, the weighted run has a weight of 2 for the need attribute (giving the need attribute roughly 4 times its unweighted influence), and weights of approximately $1/\sqrt{3}$ for each of the DSS-advantage attributes (giving each of these three attributes roughly 1/3 of its unweighted influence).

Unweighted Run

INPUT WEIGHTS
? 1,1,1,1,1,1,1,1,1

MODEL	DISTANCE
1	490
2	290
3	166
4	290
5	178
6	169
7	333
8	333
9	308
10	408
11	308
12	174
13	174
14	178
15	203
16	178
17	278
18	178
19	178
20	178
21	178
22	308
23	163
24	129
25	133
26	142
27	190
28	290
29	190
30	164
31	198
32	169
33	125
34	125
35	125
36	129
37	129
38	257
39	71
40	196
41	56
42	141
43	4
44	169
45	124

Weighted Run

INPUT WEIGHTS
? 2,1,1,1,1,1,.577,.577,.57

MODEL	DISTANCE
1	421.93
2	221.93
3	113.939
4	121.93
5	73.9296
6	114.938
7	264.93
8	264.93
9	239.93
10	339.93
11	239.93
12	57.9296
13	57.9296
14	73.9296
15	98.9296
16	73.9296
17	173.93
18	123.938
19	123.938
20	123.938
21	123.938
22	139.93
23	124.948
24	59.6246
25	75.6246
26	94.6246
27	121.93
28	221.93
29	121.93
30	74.6246
31	50.6246
32	114.938
33	41.6161
34	41.6161
35	41.6161
36	57.6161
37	57.6161
38	166.938
39	266.938
40	141.938
41	114
42	189
43	16
44	114.938
45	99.3561

In the weighted run model 43 again ranks highest, followed by model 30, then 33, 34 and 35, models 36 and 37, model 31, models 12 and 13 (assignment and transportation algorithms), and model 24.

The unweighted and weighted rankings are similar.

8. CONCLUSIONS AND RECOMMENDATIONS

For the assumed values of the attributes, the results indicate the most promising models are

43. Minimax location interactive algorithm

30. Decision tree model

31. Generalized decision tree model

33,34,35,36,37. Inventory models

24. PERT/CPM

Other models of interest are

41. Multicommodity flow heuristic

12,13. Assignment and transportation algorithms

Although it is not reflected in the estimated values of attributes, model 31 is simply an improved version of 30, and ranks below 30 only because it is not now used in the Army. Models 30 and 31 should be considered as a single model.

Adequate estimation of the attributes is beyond the scope of this survey, although it is hoped most of the attributes are of approximately the correct magnitudes. Clearly the Army need attribute is inadequately estimated. In fact, it contains no measure of whether a solution to any problem is more valuable than a solution to another problem.

Until a more accurate set of Army needs can be formulated, the recommendations—to give first consideration to minimax location, generalized decision trees, inventory models and PERT/CPM—must be

regarded as preliminary.

The program given in Appendix B can be implemented trivially on practically any computer system, and is available now at AIRMICS. It is recommended that Army personnel experiment with various attribute estimates and weightings to achieve a more reliable identification of promising models.

STATEMENT OF WORK
TCN: 78-373

SCIENTIFIC SERVICES PROGRAM

STAS

1. General

The Army Institute for Research in Management Information and Computer Science has a requirement for the services of an Operations Research Scientist to provide a brief usability survey of operations research applications software for decision support systems.

2. Objectives

The work is to provide an evaluative classification of operations research applications software-encompassing optimization, simulation and queueing software-evaluated as to their demonstrated and potential usefulness in minicomputer-based decision support systems (DSS). (A DSS is a system that provides interactive response in interfacing with a data base to aid a non-programming individual in unstructured problem solving.)

3. Specific Tasks

a. Classify operations research applications software as to data requirements, flexibility, compatibility with interactive requirements, and other dimensions affecting use in DSS.

b. Review the interactive-computing history of each class, including proposals, usage histories, and formal or informal evaluations by users.

c. Recommend a ranking procedure for selecting algorithms according to potential governmental DSS usefulness.

d. Apply the ranking procedure to recommend a small set of operations research algorithms for possible pilot implementation.

4. Reporting Requirements

A report will be submitted by 31 December 1978, written and organized so as to be suitable for use in deciding which algorithms or procedures show greatest promise for use as transformational modules in DSS.

5. Special Qualifications

The Operations Research Scientist selected for these services must be at the Ph.D. level with broad and extensive background in both deterministic and probabilistic operations research, and with a background in interactive computing.

6. Place and Period of Performance

a. Ten working days during the period 15 June 1978 to 1 October 1978, all of which will be in the Georgia Tech/AIRMICS vicinity.

b. No travel will be required.

7. Restrictions

There is no known potential conflict of interest associated with this work.

8. Security Clearance

This work is unclassified.

9. COTR

James M. Irwin
AIRMICS
313 Calculator Bldg
GA Institute of Technology
Atlanta, GA 30332
(404) 894-3107

APPENDIX B

Program to Calculate Distance Metrics

```
00100 DIM A(45,9)
00110 DIM B(1,9)
00120 MAT READ B
00130 REM IDEAL [illegible] OF ATTRIBUTES
00140 DATA 10,0,0,0,0,0,10,10,10
00150 MAT READ A
00160 REM ATTRIBUTES FOR MODELS 1 THRU 45
00170 DATA 6,10,0,10,0,10,5,0,2
00180 DATA 6,6,0,6,6,10,5,6,2
00190 DATA 6,0,0,6,0,0,5,0,1
00200 DATA 0,0,0,0,10,0,0,5,0,0
00210 DATA 3,0,0,0,0,0,5,0,3
00220 DATA 6,0,0,0,0,0,3,0,2
00230 DATA 6,10,0,5,3,0,5,0,3
00240 DATA 6,10,0,5,3,0,5,0,1
00250 DATA 6,10,3,0,0,0,5,0,3
00260 DATA 6,0,0,0,10,3,0,5,0,3
00270 DATA 6,10,0,0,0,0,5,6,2
00280 DATA 10,0,0,0,0,0,5,0,2
00290 DATA 10,0,0,0,0,0,5,0,1
00300 DATA 5,0,0,0,0,0,5,0,2
00310 DATA 3,0,0,5,0,0,5,0,2
00320 DATA 3,0,0,0,0,0,5,0,3
00330 DATA 3,0,0,10,0,0,5,0,1
00340 DATA 6,0,3,0,0,0,8,0,3
00350 DATA 6,6,3,0,0,0,6,0,2
00360 DATA 6,0,3,0,0,0,8,0,2
00370 DATA 6,0,3,0,0,0,8,0,3
00380 DATA 6,0,3,0,3,0,5,0,2
00390 DATA 6,0,3,0,3,0,5,0,6
00400 DATA 10,0,0,5,0,0,8,0,10
00410 DATA 3,0,0,5,0,0,6,0,10
00420 DATA 3,0,3,5,0,0,8,0,10
00430 DATA 6,0,0,0,0,0,5,0,2
00440 DATA 6,10,0,0,0,0,5,0,2
00450 DATA 6,0,0,0,0,0,5,0,3
00460 DATA 10,0,0,0,0,0,6,0,10
00470 DATA 8,0,0,0,0,0,8,0,10
00480 DATA 6,0,0,6,0,0,8,0,2
00490 DATA 10,0,0,0,0,0,5,0,10
00500 DATA 10,0,0,0,0,0,5,0,10
00510 DATA 10,0,0,0,0,0,5,0,10
00520 DATA 3,0,0,0,0,0,5,0,10
00530 DATA 3,0,0,0,0,0,5,0,10
00540 DATA 5,0,0,10,0,0,8,0,2
00550 DATA 10,0,2,10,2,10,8,0,3
00560 DATA 6,0,1,5,1,0,8,0,3
00570 DATA 6,5,0,5,0,0,10,10,10
00580 DATA 6,5,0,10,0,0,10,10,10
00590 DATA 3,0,0,0,0,0,10,10,10
00600 DATA 6,0,0,0,0,0,8,0,2
00610 DATA 6,0,0,0,0,0,8,0,2
00620 DIM W(1,9)
00630 PRINT"INPUT WEIGHTS"
00640 MAT INPUT W
00650 REM CALCULATE DISTANCES
00660 DIM D(45)
00670 FOR J=1 TO 45
00680 D=0
00690 FOR I=1 TO 9
00700 REM ITH ATTRIBUTE, JTH MODEL
00710 W=W(1,I)
00720 D=D+W*W*(A(J,I)-B(1,I))**2
00730 NEXT I
00740 D(J)=D
00750 NEXT J
00760 PRINT
00770 PRINT"MODEL","DISTANCE"
00780 PRINT
00790 FOR J=1 TO 45
00800 PRINT J,D(J)
00810 NEXT J
00820 END
```

END

DATE FILMED

10-81

DTIC